

Enseñanza y aprendizaje de programación. Hacia un estado del arte

Teaching and learning programming. Towards the state of the art

Gustavo Astudillo, Silvia Bast

Universidad Nacional de La Pampa, Argentina

E-mail: astudillo.gustavo@gmail.com; silviabast@exactas.unlpam.edu.ar

Resumen

A nivel mundial se tiende a incluir el aprendizaje de las ciencias de la computación, y la programación en particular, en instancias cada vez más tempranas de la educación formal. Argentina no es ajena a este proceso. En este contexto, ¿cuáles son las propuestas a nivel nacional que se enfocan en la enseñanza y el aprendizaje de la programación? ¿Cuál es el modelo pedagógico que las sustenta? ¿Qué herramientas utilizan? ¿Qué impacto ha tenido su implementación? Las respuestas a estas preguntas permiten avanzar en la configuración de un estado del arte sobre la enseñanza y el aprendizaje de la programación en Argentina. Este trabajo presenta los primeros resultados obtenidos a partir de la revisión sistemática de 49 publicaciones sobre la temática, así como la definición de un conjunto de criterios de análisis y su aplicación a las propuestas seleccionadas.

Palabras clave: programación de computadoras; enseñanza; aprendizaje; estado del arte.

Abstract

There is a worldwide tendency to include the learning of Computer Science, and particularly programming, at increasingly early stages of formal education. Argentina is not the exception. In this context, what are the proposals focused on teaching and learning programming at the national level? What is the pedagogical model that supports them? Which tools do they use? What has been the impact of their implementation? The answers to these questions allow us to define the state of the art on the teaching and learning of programming in Argentina. This article presents the first results obtained from a systematic review of 49 publications on the subject, the definition of a set of analysis criteria and their application to the selected proposals.

Keyword: computer programing; teaching; learning; state of the art.

Fecha de recepción: Septiembre 2019 • Aceptado: Noviembre 2019

ASTUDILLO, G. Y BAST, S. (2020). Enseñanza y aprendizaje de programación. Hacia un estado del arte *Virtualidad, Educación y Ciencia*, 20 (11), pp. 138-155.

Introducción

A nivel mundial existe una tendencia¹ que aboga por la inclusión de las ciencias de la computación en instancias cada vez más tempranas de la educación formal. La Argentina no es la excepción y una muestra de esto es que, desde el gobierno, se vienen impulsando distintas normas que apoyan el proceso. En 2006, en la Ley de Educación Nacional (Ley N. 26.206, 2006) se afirma que uno de los fines y objetivos de la política educativa nacional será “Desarrollar las competencias necesarias para el manejo de los nuevos lenguajes producidos por las tecnologías de la información y la comunicación.” (art. 11) y que el “acceso y dominio de las tecnologías de la información y la comunicación formarán parte de los contenidos curriculares indispensables para la inclusión en la sociedad del conocimiento.” (art. 88). En 2010, se crea el programa Conectar Igualdad que proporcionó una computadora a alumnas, alumnos y docentes de la educación secundaria pública (Decreto N. 459, 2010). En 2015, el Consejo Federal de Educación establece “que la enseñanza y el aprendizaje de la ‘Programación’ es de importancia estratégica en el Sistema Educativo Nacional” (Resolución CFE N. 263, 2015, art. 1). En 2017, se lanza el Plan Nacional Integral de Educación Digital (PLANIED) (Resolución N. 1536-E, 2017), con la misión de “integrar la comunidad educativa en la cultura digital, favoreciendo la innovación pedagógica, la calidad educativa y la inclusión socio educativa” (art. 2). Como corolario de este proceso, en 2018, se crean los Núcleos de Aprendizaje Prioritarios (NAP) para educación digital, programación y robótica, con el fin de “facilitar la integración del acceso y dominio de las tecnologías de la información y la comunicación en los contenidos curriculares indispensables para la inclusión en la sociedad digital” (Resolución CFE N. 343, 2018: 1). Esto implica la incorporación efectiva de la temática al currículum.

A lo anterior se deben sumar, sin lugar a duda, las iniciativas de la Fundación Sadosky que promueve “el estudio de la programación en las escuelas argentinas y contribuir al incremento de la matrícula en carreras relacionadas con las tecnologías de la información y las comunicaciones (TIC). Para ello lleva adelante diversos programas, entre estos, Vocaciones en TIC y Program.ar.” (Dapozo et al., 2016).

En este marco, como afirma Cobo Romani, es “fundamental desarrollar habilidades a prueba de futuro, no limitadas a ciertas herramientas, instrumentos o metodologías y adaptables a diferentes contextos y actualizables durante el aprendizaje con otros.” (2019:30). Por su parte, “la escuela, encargada de alfabetizar a la ciudadanía en saberes socialmente válidos, debe acercar a los estudiantes a la comprensión del mundo digital que nos rodea” (Echeveste & Martínez, 2016: 35).

Es así, que desde el grupo de investigación GrIDIE² se puso el foco en identificar estrategias o propuestas, a nivel nacional, que faciliten el proceso de enseñanza y de aprendizaje de la programación y que además motiven a los estudiantes. Se formularon las siguientes preguntas que guían la investigación: ¿cuáles son esas propuestas a nivel nacional?; ¿cuál es la propuesta didáctica que las sustenta?; ¿qué herramientas utilizan?; ¿qué impacto ha tenido su implementación? Las respuestas a estas preguntas conducen a la configuración de un estado del arte sobre las propuestas de enseñanza y aprendizaje de la programación en Argentina. Asimismo, con el objetivo de analizar y clasificar dichas

1 Por ejemplo: CS for All (<https://www.csforall.org/>), CS for All Teachers (<https://csforallteachers.org/>) o Developing Computational Thinking in Compulsory Education (<http://publications.jrc.ec.europa.eu/repository/>).

2 GrIDIE: Grupo de Investigación y Desarrollo en Innovación Educativa. Departamento de Matemática. Facultad de Ciencias Exactas y Naturales. Universidad Nacional de La Pampa.

propuestas se definieron un conjunto de criterios de análisis.

Este trabajo presenta los primeros resultados obtenidos a partir de la revisión sistemática de 49 artículos y posters sobre la temática, y la aplicación de los criterios a las propuestas seleccionadas.

El artículo se organiza de la siguiente manera: se presenta el marco teórico; a continuación, se detallan la metodología usada para llevar a cabo la revisión sistemática de los trabajos y los criterios de análisis; luego, se presentan los resultados a los que arribó el equipo de investigación y, finalmente, las conclusiones y trabajos futuros.

Marco teórico

En esta sección se abordan, brevemente, algunos de los conceptos que sustentan las distintas propuestas analizadas en este artículo y permitirán dar un marco de referencia para el abordaje de la sección de resultados.

Pensamiento computacional

Término acuñado por Jannette Wing quien, hace más de 10 años, afirmaba que “será una habilidad y una actitud de aplicación universal para todas las personas” (Wing, 2006). La autora define el pensamiento computacional (PC) como “el proceso de pensamiento involucrado en la formulación de un problema y la expresión de su(s) solución(es) de tal manera que una computadora, humana o mecánica, pueda llevarla(s) a cabo efectivamente” (Wing, 2017: 8).

Si bien son variadas las estrategias que permiten desarrollar el PC, los investigadores coinciden en que las ciencias de la computación (CC), y la programación en particular, propician el desarrollo de las habilidades que son propias del PC. El “aprendizaje de lenguajes de programación y la creación de algoritmos permite desarrollar lo que se denomina pensamiento computacional” (Zabala, Cerrato, Blanco, Morán, & Teragni, 2016: 114).

Según Zapata Ros, algunas habilidades propias del pensamiento computacional que se busca estimular en los estudiantes son “el análisis ascendente, [...] el pensamiento divergente o lateral, la creatividad, la resolución de problemas, el pensamiento abstracto, la recursividad, la iteración, los métodos por aproximaciones sucesivas, el ensayo-error” (2015: 11).

El objetivo del Pensamiento Computacional es desarrollar sistemáticamente las habilidades de pensamiento crítico y resolución de problemas con base en los conceptos de la computación. (Dapozo et al., 2016: 114).

La enseñanza de la programación

La década de los 80' se caracterizó por el uso, en el contexto educativo, de los lenguajes BASIC y LOGO. En los 90', “se consideraba fuertemente a la educación como capital de consumo y privatización” (Martínez & Echeveste, 2015: 4) y la enseñanza de computación en las escuelas tuvo una fuerte concepción operativa e instrumental, enfocándose en la enseñanza de software utilitario en detrimento de los conceptos propios de CC (Levis, 2007). Con la sanción, en 2006, de la Ley de Educación Nacional (Ley N. 26.206, 2006), se plantea la enseñanza de las tecnologías de la información

y la comunicación (TIC) y que algunas escuelas tengan la orientación en informática, esto vuelve a poner el foco en la enseñanza de la programación, aunque de forma aislada y muy incipiente.

“La enseñanza de la computación, más específicamente de la programación, presenta particularidades que desafían [al] régimen académico estándar” (Echeveste & Martínez, 2016: 35). Programar implica el “uso simultáneo de la creatividad, un conjunto de conocimientos técnicos asociados y la aptitud de trabajar con abstracciones, tanto simbólicas como mentales.” (Rosas, Zúñiga, Fernández, & Guerrero, 2017: 697). La programación no se trata solo de escribir un programa, sino que involucra “aprender a entender un problema, a plantear soluciones efectivas, a manejar lenguajes para expresar una solución, a utilizar herramientas que entiendan esos lenguajes, a probar que la solución sea válida, a justificar las decisiones tomadas” (Dapozo, Greiner, Petris, Espíndola, & Company, 2017: 1122), entre otras, y por tanto el proceso de enseñanza debe contemplar estos aspectos.

Actualmente, “tanto la programación como el pensamiento computacional resultan relevantes para el aprendizaje” (Ripani, 2017: 15) y se ha corrido el foco de los lenguajes de programación, que ya “no son fines en sí mismos, son herramientas. Los recursos didácticos existen, los docentes elegirán el que mejor se adapte a sus circunstancias” (Piccin & Cicinelli, 2017: 706).

Entre las estrategias utilizadas para que los lenguajes de programación dejen el centro de la propuesta didáctica, puede mencionarse el uso de los entornos de programación por bloques, como Scratch, Alice o Pilas Bloques, entre otros. En ellos, se construyen programas encastrando bloques en la secuencia correcta para lograr el objetivo planteado. Tales entornos, frecuentes en las propuestas de enseñanza de la programación, presentan ventajas como la ausencia de errores de sintaxis propia de los lenguajes textuales y la representación visual de conceptos abstractos.

Robótica educativa

A partir de la reducción de los costos y del hardware libre, una estrategia que cobra fuerza es el uso de dispositivos tecnológicos que colaboran con el aprendizaje y la motivación de estudiantes, entre los que se encuentran los denominados robots educativos. Estos dispositivos electrónicos pueden ser controlados por medio de programación e interactúan con el entorno a través de distintos tipos de sensores.

Estos robots son la conexión ideal entre la programación con una impronta lúdica y la representación de las instrucciones sobre un contexto real. Como afirma Vaillant, “La robótica forma parte de un enfoque pedagógico centrado en el alumno, que le permite construir objetos tangibles de su propio diseño y con sentido para él” (2013: 38). Para Monsalves (2011) y Ruiz-Velasco (2007) se trata de una disciplina que tiene como objetivo “generar entornos de aprendizaje heurísticos” poniendo el foco en la participación activa, donde los aprendizajes se construyen a partir de la experiencia del estudiante durante el proceso de armado y programación de los robots.

Inicialmente, parecería ser un aporte para el desarrollo del pensamiento computacional, en diversas áreas curriculares. Como afirma Gritis y Vázquez, “la robótica se convierte en un apoyo para el aprendizaje. En este sentido, el uso de estos objetos tecnológicos puede ayudar de una manera distinta a abordar contenidos curriculares, aprovechando las diversas posibilidades que este

tipo de tecnología puede brindar” (2014: 145-146). Específicamente en el área de la informática, se relaciona con la posibilidad de facilitar el aprendizaje de un lenguaje de programación, propiciar la experimentación y estimular las competencias asociadas a la resolución de problemas.

Juegos serios

Jugar ha sido desde siempre una actividad informal de aprendizaje. En el ámbito digital, las consolas, las salas de videojuegos, las computadoras y, en los últimos años, los celulares y las tabletas han permitido la incorporación de los videojuegos a la vida cotidiana de la gente.

El ámbito educativo es permeable también a esas influencias, por lo que, explorar el potencial educativo de animaciones interactivas y juegos digitales da a las instituciones escolares, y a docentes en particular, la oportunidad para incluir estrategias en las que el aprendizaje resulte divertido y atractivo. Debe tenerse en cuenta también que los estudiantes que asisten a las aulas en la actualidad forman parte de una “generación de pantalla y juegos” y se caracterizan por estar expuestos a un bombardeo constante de estímulos multimediales y multisensoriales, tienen una necesidad permanente de novedades para evitar el aburrimiento, cuentan con la ubicuidad de las pantallas, exigen la respuesta instantánea a acciones y las respuestas como reacción, entre otras características (Prensky, 2007; Aldrich, 2009).

Según ProActive (2010), los videojuegos presentan conflictos, objetivos, desafíos y reglas que incentivan y atrapan al jugador; ciclos de retroalimentación cortos que permiten al jugador percibir rápidamente el impacto y las consecuencias de sus acciones. Proponen inmersión y compromiso a través de historias atractivas en mundos virtuales. Cuentan con un sistema de recompensas que invita a los jugadores a hacer sus mejores esfuerzos. Estimulan la competencia y la colaboración. Otro aspecto importante es que tratan el error como parte del proceso de aprendizaje. “Los juegos deben tratar los errores y lograr que los mismos sean una oportunidad para proporcionar retroalimentación y estimular al jugador” (Kapp, 2012: 99). Estas características los convierten en materiales adecuados para desarrollar actividades que intenten promover un aprendizaje significativo.

Las investigaciones sobre videojuegos han llevado a desarrollar clasificaciones y estrategias de su uso en el aula. Así es posible encontrar juegos serios, videojuegos que se usan para educar, entrenar e informar (Michael & Chen, 2005). Estrategias como Games-Based Learning que “refiere a la integración de juegos o la mecánica de los juegos dentro de experiencias educativas” (Johnson et al., 2012: 19).

Gamificación

La gamificación (también denominada ludificación, o en inglés gamification), hace referencia al “uso de elementos del diseño de juegos en contextos no lúdicos” (Deterding, Khaled, Nacke, & Dixon, 2011: 2). Esto significa aplicar metáforas que se corresponden con los juegos a otros contextos, como el educativo, en busca de mejorar la motivación y aumentar la participación. Como afirman Gallego et al., “gamificar es plantear un proceso de cualquier índole como si fuera un juego. Los participantes son jugadores y como tales son el centro del juego” (2014: 2).

“La gamificación no es utilizar juegos para la enseñanza de contenidos” (Vera, Rodríguez, &

Moreno, 2018: 2), como afirma Kapp (2012), es imperativo que todos los elementos que definen a este enfoque estén presentes para hacer un uso genuino de este. Así, la propuesta debe estar basada en juegos (reglas, interactividad, retroalimentación, etc.), incluir niveles, recompensas, insignias y/o puntos. También son importantes la estética y el pensamiento de juego (la competición, la exploración, la narración o el relato de una historia); contar con un objetivo explícito, la motivación y el planteo de uno o varios problemas a resolver.

La aplicación de gamificación, en el contexto educativo, va más allá de la motivación; busca promover los aprendizajes a partir de la resolución de problemas al interactuar con un ambiente de juego y/o con otros jugadores.

Metodología

Revisión sistemática

La revisión sistemática (Astudillo, Bast, Segovia, & Castro, 2019) se realizó con base en la metodología propuesta por Kitchenham (2004), en esta se propone: (a) definir preguntas de investigación, (b) trazar una estrategia de búsqueda (dónde buscar, con qué palabras claves), (c) establecer criterios de inclusión y exclusión, que serán aplicados, tanto para la selección inicial, como para la selección final.

Las preguntas de investigación que se intentan responder son: ¿cuál es la estrategia propuesta para la enseñanza de la programación?; ¿a qué nivel educativo está dirigida la propuesta?; ¿qué aprendizajes/habilidades se intentan desarrollar en los estudiantes?; ¿cuál es la secuencia didáctica?; ¿cuál es el enfoque pedagógico que sustenta la propuesta?; ¿cómo se evalúan los aprendizajes?; ¿qué recursos de software y hardware se utilizan para implementar las propuestas?; ¿qué resultados se obtuvieron en la implementación de las propuestas didácticas?

La búsqueda se focalizó en JADiPro18, WICC (17/18), TE&ET (17/18) y CONAISI (17/18); y las revistas TE&ET (17/18) y VEC (17/18). Además, se sumaron artículos que se recuperaron a partir de las referencias de las publicaciones seleccionadas.

La búsqueda se hizo en base a la combinación de palabras como: “programación”, “pensamiento computacional”, “enseñanza”, “aprendizaje”.

Las publicaciones incluidas fueron aquellas que verificaron, de manera excluyente, los siguientes criterios: propuestas que abordan estrategias para la enseñanza de la programación en nivel secundario y/o universitario, que se diseñan/implementan para Argentina, escritos en castellano o inglés, y accesibles de texto completo.

El proceso preliminar se inició con la revisión de artículos que contuviesen las cadenas de búsqueda en título, palabras claves o resumen. Leído el resumen, se aplicaron los criterios de inclusión y exclusión. Finalmente, se comenzó con la lectura completa de los artículos en busca de responder las preguntas de investigación. Luego de estas lecturas, se volvieron a aplicar los criterios de inclusión y exclusión.

Criterios de análisis

En busca de caracterizar y clasificar las propuestas, se definieron 18 criterios de análisis, los cuales son presentados a continuación:

1. Autores: se registran los autores con el fin de identificar las propuestas que se desarrollan en distintas publicaciones.
2. Institución/es: se registra universidad y unidad académica (UA), en busca de identificar las distintas universidades que trabajan en la temática.
3. Localización de la institución: se registran las coordenadas de la UA con el fin de construir un mapa interactivo de propuestas.
4. Año de publicación: se registra el año de la publicación con el fin de tener una evolución, por año, de las propuestas. Tanto en cómo evoluciona una propuesta, así como en la cantidad que hay por año.
5. Palabras clave: se registran las palabras clave del artículo o en su defecto, si no las tuviere, del repositorio que lo aloja. Este registro permite determinar, a través de una gráfica de frecuencias, las temáticas sobre las que se está trabajando y alimentar o redefinir las palabras usadas para la búsqueda.
6. Fuente: se registra dónde se publicó el artículo, con el objetivo de registrar las reuniones científicas o revistas en las que se están publicando e identificar nuevas fuentes de búsqueda.
7. Clasificación: se hace una clasificación de la propuesta. Se toma como base una clasificación inicial a partir de las primeras lecturas (robótica educativa, gamificación, juegos serios y aprendizaje móvil), se registra si se puede agrupar bajo uno de estas clasificaciones (o más) o si se debe crear una nueva. El objetivo es definir una forma de clasificación/agrupación de las propuestas.
8. Destinatarios: se registra si la propuesta se enfoca en el nivel medio, el terciario o el universitario (se distingue las que apuntan al ingreso universitario), o en formación docente (o alguna combinación de estos).
9. Enfoque pedagógico: en este criterio se registran frases/palabras que permitan identificar el modelo pedagógico que subyace a la propuesta.
10. Materiales didácticos: se registran aquí datos que permitan dar cuenta del material educativo, producido en el marco de la propuesta, que la acompañe.
11. Habilidades/conceptos: aquí se registran habilidades, conceptos, competencias que se intentan desarrollar/potenciar, en los destinatarios, a través de cada propuesta.
12. Secuencia/Itinerario de aprendizaje: se registra la secuencia de actividades que se lleva adelante en cada propuesta, para que los participantes alcancen las habilidades o conceptos expuestos.
13. Lenguaje de programación: se registra el lenguaje de programación que se utiliza en la propuesta, sea o no el foco del aprendizaje o la herramienta para alcanzar las habilidades/conceptos propuestos.
14. ¿Se aplicó?: aquí se da cuenta de la aplicación de la propuesta.

15. Herramientas: se registran las aplicaciones que se utilizan como recurso didáctico en el marco de la propuesta.
16. Recursos: este criterio permite registrar otros recursos que se utilicen en la propuesta (por ejemplo, robots).
17. Evaluación de los aprendizajes: se describen todas las instancias de evaluación establecidas en la propuesta.
18. Resultados obtenidos: se describen aquí los resultados obtenidos al llevar a cabo la propuesta.

Cabe aclarar que si el criterio no aplica el atributo se deja en blanco, pero no se descarta la publicación.

Resultados

Durante el proceso preliminar se seleccionaron 115 publicaciones, de las cuales 34 fueron posters. Se seleccionaron 49 producciones en el proceso de selección final³. De estas, la mayoría (17) pertenece a workshops, le siguen artículos presentados en congresos (16); del resto, 10 fueron publicados en revistas; 5, en jornadas, y 1, en una conferencia (tabla 1).

Tabla 1. Distribución de las producciones analizadas por evento

Tipo de evento	Cantidad	Nombre
Workshops	17	Workshop de Investigadores en Ciencia de la Computación (WICC)
Congresos	16	Congreso Argentino de Ciencias de la Computación (CACIC) Congreso Nacional de Ingeniería Informática – Sistemas de Información (CONAIISI) Congreso de tecnología en educación y educación en tecnología (TE&ET)
Revistas	10	Campus Virtuales (CV) Revista de Educación a Distancia (RED) Revista Iberoamericana De Tecnología En Educación Y Educación En Tecnología (TE&ET) Virtualidad, Educación y Ciencia (VEC)
Jornadas	5	Jornadas Argentinas de Didáctica de la Programación (JADiPro) Jornadas Argentinas de Informática (JAIIO) Jornadas sobre la Enseñanza Universitaria de la Informática (JENU)
Conferencia	1	Latin American Computing Conference (LACC), organizada por el Centro Latinoamericano de Estudios de Informática (CLEI)

Fuente: elaboración propia

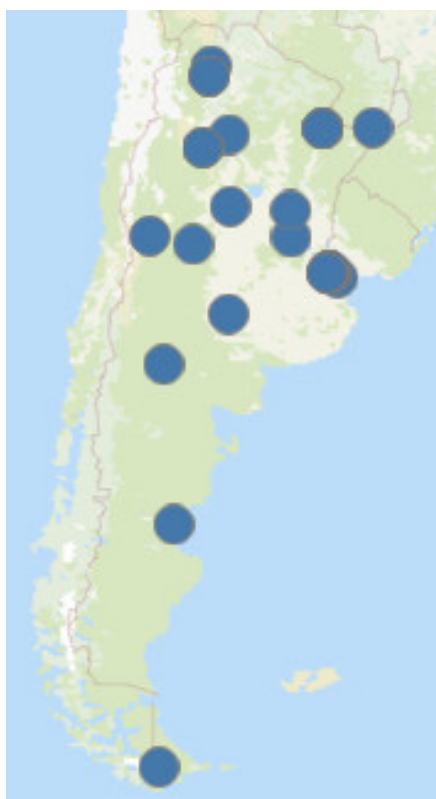
³ Cabe aclarar que este número indica un corte en el proceso para el presente artículo y que el proceso de selección final continúa.

Una vez tabuladas todas las publicaciones fue posible realizar su análisis con base en los criterios definidos. A continuación, se presentan los primeros resultados.

De las 49 publicaciones analizadas se obtuvieron 32 propuestas que abordan diferentes aspectos de la enseñanza de la programación. Estos van desde enfoques más generales sobre didáctica de la programación, hasta secuencias didácticas que permiten desarrollar el pensamiento computacional o la apropiación de conceptos de la programación.

Desde el punto de vista geográfico, hay 16 provincias representadas a través de 23 unidades académicas (figura 1). Desde lo demográfico, son 139 investigadores dedicados al tema, 65 mujeres y 74 varones.

Figura 1: Distribución de las propuestas en Argentina

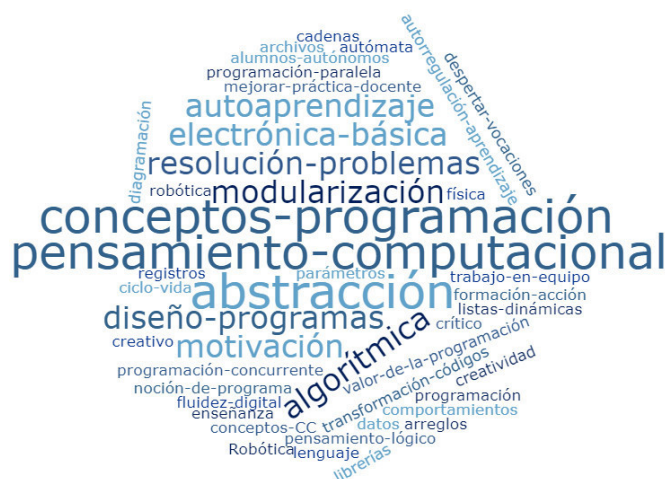


Fuente: elaboración propia, generada con Qlik Sense Cloud.

Del análisis de las palabras clave, extraídas de cada publicación⁴, se pudieron identificar como las de mayor frecuencia: programación, enseñanza, pensamiento computacional y robótica educativa. En la figura 2, se pueden observar dos nubes de palabras, la primera (figura 2.a) en la que se trabajó con las palabras/sustantivos (excluyendo conectores entre palabras); y en la segunda (figura 2.b), se trabajó uniando las palabras con guiones y unificando las similares (como, por ejemplo, singular y plural de la misma palabra).

4 Cabe aclarar que en caso de no contar la publicación con palabras clave se tomaron las del repositorio que la aloja.

Figura 3: Distribución de frecuencia de “Habilidades/conceptos”



Fuente: Elaboración propia, generada con nubedepalabras.es.

La mayoría de las propuestas (27/32) tuvieron su correlato en alguna experiencia de campo (criterio: “¿Se aplicó?”). Se implementaron a partir de talleres en escuelas, para ingresar a la universidad y cátedras de los primeros años.

En relación con la “secuencia de aprendizaje” que se expone en los trabajos, si bien el análisis se encuentra en un estado incipiente y requiere de profundización, pueden abstraerse algunas características, y de acuerdo con los enfoques que predominan en cada una de ellas, podrían clasificarse en: robótica educativa, gamificación y juegos serios (criterio: “clasificación”).

En cuanto a las propuestas que se clasificaron dentro de robótica educativa (Blas, Castellaro, Mandracchia, & Hauque, 2017; de la Fuente et al., 2018), solo 5 describen explícitamente la secuencia que aplican. En todos los casos se menciona la motivación de los estudiantes y las ventajas que ofrece la robótica para ver físicamente los resultados de la ejecución de un programa y hacen énfasis en la mejora iterativa del código, a partir de la observación de las acciones ejecutadas por el robot (Zurita et al., 2017). En general, al inicio de la secuencia hay un acercamiento a la programación a través de herramientas como RoboMind o el desarrollo RITA y continúan con lenguajes como Arduino, Python y Java en etapas posteriores.

Las propuestas enfocadas en gamificación (Astudillo, Bast, & Willging, 2016; Vera et al., 2017) presentan secuencias de aprendizajes; en general, plantean desafíos que los estudiantes deben superar para pasar de nivel y asignan puntos e insignias. Se incorporan en esta clasificación las publicaciones que utilizan jueces en línea (Ambort, Daniel & Castellaro, Marta, 2016; Kogan, Rodríguez, & Amigone, 2017) que, si bien no mencionan la gamificación, hacen uso de desafíos, ranking, puntajes para incentivar a los participantes, por eso se incluyen en esta categoría.

Las propuestas relacionadas con juegos serios (Ambort, Daniel & Castellaro, Marta, 2016; Garis, Ana, Albornoz, Claudia, & Silvestri, Mario, 2017), en general, usan aplicaciones tales como Code.org, LightBot (Garis, Albornoz, & Silvestri, 2018), para continuar luego con aplicaciones que permiten programar con bloques, como Scratch (Dapozo et al., 2016; Casali, Zanarini, San Martín, & Monjela,

2018). En algunos casos, se desarrollan herramientas propias (Bione, Miceli, Sanz, & Artola, 2017; Martínez López, Ciolek, Arévalo, & Pari, 2017) para introducir los conceptos de programación.

Refutando la hipótesis inicial (Astudillo, Bast, Willging, et al., 2019), son pocos los trabajos que incluyen aprendizaje móvil. Los que lo hacen proponen el desarrollo de aplicaciones simples en dispositivos móviles, usando herramientas tales como OpenDiagramar o MIT App Inventor (Queiruga et al., 2017; Reyes et al., 2017).

Un conjunto de propuestas, clasificadas como “otras”, hacen uso de realidad aumentada (Salazar Mesía, Sanz, & Gorga, 2016), video educativo (Torres, García, Lamas, & Pérez Ibarra, 2017), clase invertida (Prado & Lara, 2018) y actividades unplugged (Rosas, Zúñiga, Fernández, & Guerrero, 2018).

Al analizar los “lenguajes de programación” que se mencionan en los trabajos, no se detecta una tendencia muy marcada. Puede observarse una predominancia de C++, Python y C, seguidos por Java, Pascal y Arduino. En la Figura 4 se muestra una nube de palabras generada a partir de la frecuencia con que los lenguajes son mencionados en las publicaciones.

Figura 4: Distribución de frecuencia de “Lenguajes”



Fuente: Elaboración propia, generada con nubedepalabras.es.

En las producciones enfocadas en la robótica educativa, se utilizan “herramientas” como: Dropsy, XRemoteBot, DuinoBotSocks que son desarrollos propios, precursores de otras aplicaciones, la plataforma web FrankLab, Blokly, Lego Mindstorm, Easy Duino, RoboMind.

Las herramientas usadas en los trabajos en los que predomina la gamificación mencionan sitios del tipo Jueces en línea, como URI Online Judge, Hornero, plataformas como CoDin Game o Project Euler.net. También se hace uso de aplicaciones como RITA, Code.org y Scratch incluyendo las mencionadas aplicaciones como parte de la propuesta gamificada.

En cuanto a las herramientas que usan en los trabajos, que caen dentro de la clasificación de Juegos Serios, predominan Code.org, seguido por LightBot, aparecen luego aplicaciones como RITA, Gobstones, Astrocódigo, y una API para Minecraft. Cabe destacar que en muchos casos amplían la

propuesta incluyendo herramientas de programación en bloques como Scratch, Pilas Bloques, Pilas Engine y Alice.

Con respecto a los “recursos” que se utilizaron en las propuestas, en general con utilización de robótica educativa, fueron principalmente placas Arduino, sensores (distancia, colores, entre otros) actuadores (LED, motores DC, buzzer, entre otros) y robots (Múltiplo N6 y Lego Mindstorms). También se registró el uso de teléfonos inteligentes y tabletas en los enfocados en aprendizaje móvil.

En un análisis preliminar, al hacer foco en los “resultados” obtenidos de la aplicación de las propuestas en el aula (19/32), puede notarse que estos plantean una mejora en la apropiación de los conceptos (11/19) reflejada, en general, en un mejor rendimiento. También, que las propuestas aumentan la motivación (9/19) en la resolución de las actividades.

Dentro de los resultados de la investigación, por fuera de la clasificación inicial, fue posible identificar publicaciones enfocadas fuertemente en lo pedagógico/didáctico. Estas se posicionan tanto desde la enseñanza como desde el aprendizaje y dan cuenta de aspectos a tener en cuenta al momento de diseñar una propuesta:

- Piccin & Cicinelli (2017) proponen un framework con las siguientes características: a) Un modelo conceptual de la enseñanza de la programación, b) un conjunto de requerimientos para la construcción de herramientas, c) recomendaciones para el uso de dichas herramientas asociadas a la modalidad de enseñanza preferida por el docente y las competencias o contenidos en cuestión.
- Echeveste & Martínez (2016: 35) pudieron identificar cuatro situaciones a tener en cuenta al trabajar los conceptos de programación en el aula: “la inclusión de jóvenes considerados ‘problemas’, el saber práctico de la programación, el tiempo prolongado que lleva la programación y el trabajo heterogéneo y grupal.”.
- Dapozo et al. (2016: 121) realizan una propuesta centrada en el aprendizaje basado en problemas, para desarrollar la “abstracción”, trabajando sobre la “descomposición del problema en partes, traducida a la creación de bloques en la solución diseñada y en la legibilidad de la solución, mediante la definición de nombres representativos a los bloques”.
- Rosas et al. (2018) realizan una propuesta basada en el aprendizaje por descubrimiento, la resolución de problemas y la computación unplugged para desarrollar el PC.

Finalmente, parece interesante enfocarse o mirar el impacto de la producción de software educativo por el propio equipo docente. Como afirma Spigariol:

Que los docentes de programación puedan utilizar como recurso pedagógico un software creado por ellos mismos es una singularidad que permite un sinnúmero de posibilidades, como aporte e incentivo a la investigación y por la potencialidad de ir adaptando permanentemente la herramienta a las nuevas necesidades que descubren y opciones que toman (2015: 1231)

El hecho de que el mismo equipo docente sea quien más profundamente conoce los requisitos y características que debe tener la herramienta y también ser su desarrollador soluciona el problema comunicacional que se presenta generalmente entre los clientes y el equipo de desarrollo. Además, una vez que el software se encuentra operativo, se obtiene una retroalimentación rápida y de primera mano de parte de los estudiantes que hacen uso de este, lo que posibilita la mejora iterativa de la

aplicación hasta que logre un punto de estabilidad o hasta que surjan nuevas necesidades.

Como se ha mencionado anteriormente, entre los recursos didácticos de los trabajos, aparecen aplicaciones que son desarrollos propios de los equipos de investigación, tales como: Wollok (Spigariol, 2015), RITA (Aybar Rosales, Queiruga, Banchoff Tzancoff, Kimura, & Brown Bartneche, 2017), Hornero (Kogan et al., 2017), Gobstones (Martínez López et al., 2017), plataforma web de gamificación para aprendizaje de C (Vera et al., 2018), FrankLab (de la Fuente et al., 2018), DROPSY, XRemoteBot y DuinoBotSocks (Queiruga et al., 2017), Davinci Concurrente (Depetris et al., 2018) y Astrocódigo (Bione et al., 2017).

Conclusiones y trabajos futuros

En este artículo se presentaron un conjunto de criterios y su aplicación sobre 32 propuestas de enseñanza y/o aprendizaje de la programación. Dicho análisis, aunque con resultados preliminares, permitió caracterizar y clasificar las propuestas analizadas.

La revisión de las propuestas permitió trazar un mapa a nivel nacional sobre la investigación en la temática e identificar un notable interés en virtud de la evolución de distintas propuestas a lo largo de los años analizados. Así como un alto nivel de su aplicación en experiencias concretas en el aula.

Fue posible revisar la clasificación inicial de las propuestas: robótica educativa, juegos serios, gamificación y aprendizaje móvil, eliminando este último y sumando enseñanza de la programación. Asimismo, surge otra posible clasificación en función de las habilidades a desarrollar.

Si bien las aplicaciones ya tradicionales en las propuestas de enseñanza y aprendizaje de la programación como Lightbot, code.org, RoboMind, Scratch, Pilas Engine o Pilas Bloques son utilizadas, también se encontró un especial interés por el desarrollo de software educativo ad hoc. Lo que podría abrir una línea de investigación acerca del impacto, tanto en los estudiantes como en el equipo docente de este tipo de actividad, que pone al profesor en este doble rol de profesional de la informática y docente del área.

Puede observarse una preferencia por el uso de la robótica educativa, seguida por el uso de juegos serios/programación por bloques y la gamificación, sin embargo, no se trata en general de enfoques puros, sino que combinan robótica educativa con juegos serios/programación por bloques o gamificación con juegos serios/programación por bloques. Además, se buscan otras estrategias como el aprendizaje móvil, la realidad aumentada y la clase invertida. Esto muestra un marcado interés por el desarrollo de propuestas innovadoras, aunque en algunos casos centradas más en un desarrollo tecnológico que en lo didáctico/pedagógico. Respecto a esto último, se puede observar también una marcada preocupación por realizar aportes concretos a la didáctica de la CC.

A futuro se continuará con el procesamiento de algunas publicaciones: selección y procesamiento de trabajos de 2019. Se pretende seguir con este proceso, a fin de constituir un observatorio/repositorio de propuestas que permita a investigadores y docentes acceder a estas. Asimismo, se espera contar con un mapa interactivo donde sea posible visibilizar las propuestas en un mapa de Argentina con acceso a información.

Del análisis de las palabras clave se espera tomar información para reorientar la búsqueda.

Asimismo, se torna indispensable profundizar el estudio desde lo pedagógico/didáctico. Analizar las secuencias de aprendizaje, la evaluación y los enfoques pedagógicos subyacentes en las propuestas analizadas. Con miras a avanzar en ese sentido, se ha incorporado recientemente al proyecto una investigadora con un perfil pedagógico.

Referencias bibliográficas

- AMBORT, D., & CASTELLARO, M. (2016). Estrategias para la motivación e integración en el primer año universitario: Innovación mediante resolución de problemas y trabajo colaborativo. *Virtualidad, Educación y Ciencia*, 7(12), 85–98.
- ASTUDILLO, G. J., BAST, S. G., SEGOVIA, D., & CASTRO, L. (2019). Revisión de propuestas para la enseñanza de la programación. Presentado en XII Congreso de Tecnología en Educación y Educación en Tecnología, San Luis, Argentina.
- ASTUDILLO, G. J., BAST, S. G., & WILLGING, P. A. (2016). Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación. *Virtualidad, Educación y Ciencia*, 7(12), 125-142.
- ASTUDILLO, G. J., BAST, S. G., WILLGING, P., SEGOVIA, D., CASTRO, L., LUCERO, P., DISTEL, J. M. (2019). Estrategias innovadoras en los procesos de enseñanza y de aprendizajes de la programación. Presentado en XXI Workshop de Investigadores en Ciencias de la Computación, San Juan, Argentina. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/77161>.
- AYBAR ROSALES, V. DEL C., QUEIRUGA, C. A., BANCHOFF TZANCOFF, C. M., KIMURA, I., & BROWN BARTNECHE, M. (2017). Programming Competitions in High School Classrooms: RITA en RED. Presentado en Simposio Latinoamericano de Informática y Sociedad (SLIS-CLEI) - JAIIO 46 (Córdoba, 2017). Recuperado de <http://sedici.unlp.edu.ar/handle/10915/63318>.
- BIONE, J., MICELI, P., SANZ, C. V., & ARTOLA, V. (2017, junio). Enseñanza de la programación con astrocódigo. Presentado en XII Congreso de Tecnología en Educación y Educación en Tecnología, La Matanza, Buenos Aires. Recuperado de <http://hdl.handle.net/10915/63453>.
- BLAS, M. J., CASTELLARO, M., MANDRACCHIA, A. D., & HAUQUE, F. (2017). Secuencia y Material Didáctico para un Primer Curso de Programación Empleando RoboMind y un Robot Móvil Arduino. *Educación en Ingeniería*, 1177-1190. Santa Fe, Argentina.
- CASALI, A., ZANARINI, D., SAN MARTÍN, P. S., & MONJELAT, N. (2018). Pensamiento computacional y programación en la formación de docentes del nivel primario. XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste).
- COBO ROMANÍ, C. (2019). *Acepto las condiciones: Usos y abusos de las tecnologías digitales*. Madrid: Fundación Santillana.
- DAPOZO, G. N., GREINER, C. L., PETRIS, R. H., ESPÍNDOLA, M. C., & COMPANY, A. M. (2017). Enseñanza de la Programación en la Universidad. Factores que Inciden en el Buen Desempeño de los Estudiantes. *Educación en Ingeniería*, 1122-1128. Santa Fe, Argentina.
- DAPOZO, G. N., PETRIS, R. H., GREINER, C. L., ESPÍNDOLA, M. C., COMPANY, A. M., & LÓPEZ, M. (2016). Capacitación en programación para incorporar el pensamiento computacional en las escuelas. *TE & ET*, 18. Recuperado de <http://hdl.handle.net/10915/58516>.

- DE LA FUENTE, J., PICUCCI, M., BONET PEINADO, D., ZURITA, R., PARRA, G., RODRÍGUEZ, J., & CECCHI, L. (2018). Construyendo franklab: Una plataforma web de robótica educativa. XXIV Congreso Argentino de Ciencias de la Computación, 1030-1039. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/73601>.
- DECRETO N. 459. Creación del «PROGRAMA CONECTAR IGUALDAD.COM.AR», Boletín Oficial del Poder Ejecutivo Nacional, Argentina, 7 de abril 2010.
- DEPETRIS, B. O., FEIERHERD, G. E., PENDENTI, H., AGUIL MALLEA, D., TEJERO, C. G., PRISCHING, G., ... MAMANI, J. (2018). Diseño y aplicación de estrategias para la enseñanza inicial de la programación. Presentado en XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste). Recuperado de <http://sedici.unlp.edu.ar/handle/10915/68652>.
- DETERDING, S., KHALED, R., NACKE, L. E., & DIXON, D. (2011). Gamification: Toward a definition. CHI 2011 Gamification Workshop Proceedings. Presentado en CHI 2011 Workshop, Vancouver, BC, Canada.
- ECHEVESTE, M. E., & MARTÍNEZ, M. C. (2016). Desafíos en la enseñanza de Ciencias de la Computación. Virtualidad, Educación y Ciencia, 7(12), 34-48.
- GALLEGO, F., MOLINA, R., & FARAÓN, L. (2014). Gamificar una propuesta docente. Diseñando experiencias positivas de aprendizaje. Presentado en XX Jornadas sobre la enseñanza universitaria de la informática, Oviedo, España. Recuperado de <http://hdl.handle.net/10045/39195>.
- GARIS, A. G., ALBORNOZ, M. C., & SILVESTRI, M. A. (2018). Analizando el impacto de talleres de programación en escuelas con respecto al ingreso de alumnos en carreras de informática. Presentado en WICC 2018. Recuperado de <http://hdl.handle.net/10915/67564>.
- GARIS, ANA, ALBORNOZ, CLAUDIA, & SILVESTRI, MARIO. (2017). La Hora del Código: Promoviendo la Programación en Escuelas Secundarias de San Luis. Actas del 5to Congreso Nacional de Ingeniería Informática / Sistemas de Información Educación en Ingeniería, 1215-1220. Recuperado de <http://tecnomate.xyz/Actas-CONAIIISI-2017.pdf>.
- KOGAN, P., RODRÍGUEZ, J., & AMIGONE, F. (2017). Agente Hornero: Ampliando las posibilidades de aprender a programar. Presentado en XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires). Recuperado de <http://hdl.handle.net/10915/62262>.
- LEVIS, D. (2007). Enseñar y aprender con informática/enseñar y aprender informática. Medios informáticos en la escuela argentina. CABELLO, R. y DS LEVIS (comps.) (2007). Medios informáticos en la educación: a principios del siglo XXI, Prometeo, Buenos Aires, 21-50.
- LEY N. 26.206. Ley de Educación Nacional. Boletín Oficial del Honorable Congreso de la Nación Argentina, Buenos Aires, Argentina, Pub. L. No. 26206, 28 de diciembre de 2006.
- MARTÍNEZ LÓPEZ, P. E., CIOLEK, D., ARÉVALO, G., & PARI, D. (2017). The GOBSTONES method for teaching computer programming. 2017 XLIII Latin American Computer Conference (CLEI), 1-9. <https://doi.org/10.1109/CLEI.2017.8226428>
- MARTÍNEZ, M. C., & ECHEVESTE, M. E. (2015). Representaciones de estudiantes de primaria y secundaria sobre las Ciencias de la Computación y su oficio. Revista de Educación a Distancia, (46). Recuperado de <https://revistas.um.es/red/article/view/240241>.
- PICCIN, A. M., & CIGINELLI, D. (2017). Framework para la construcción de estrategias didácticas para la

- enseñanza inicial de la programación de computadoras. Presentado en XIX Workshop de Investigadores en Ciencias de la Computación, Buenos Aires. Recuperado de <http://hdl.handle.net/10915/62321>
- PRADO, A. DEL, & LARA, L. R. (2018). Herramientas TIC para la enseñanza de programación, empleando aula invertida. Presentado en XIII Congreso de Tecnología en Educación y Educación en Tecnología (Posadas, 2018). Recuperado de <http://sedici.unlp.edu.ar/handle/10915/69076>.
- QUEIRUGA, C. A., BANCHOFF TZANCOFF, C. M., MARTÍN, S. S., AYBAR ROSALES, V. DEL C., LÓPEZ, F. E. M., KIMURA, I., & GÓMEZ, S. (2017). PROGRAMAR en la escuela: Nuevos desafíos en las aulas. Presentado en XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires). Recuperado de <http://hdl.handle.net/10915/62341>.
- RESOLUCIÓN CFE N. 263. Establecer que la enseñanza y el aprendizaje de la “Programación” es de importancia estratégica en el Sistema Educativo Nacional. , Pub. L. No. 263 (2015).
- RESOLUCIÓN CFE N. 343. Núcleos de Aprendizaje Prioritarios para Educación Digital, Programación y Robótica. , Pub. L. No. 343 (2018).
- RESOLUCIÓN N. 1536-E. Creación del PLAN NACIONAL INTEGRAL DE EDUCACIÓN DIGITAL (PLANIED). , Pub. L. No. 1536- E (2017).
- REYES, C. J., MASSÉ PALERMO, M. L., ESPINOZA, C. N., VARGAS, C., RAMÍREZ, J., & TRENTI, J. (2017). Dispositivos móviles como soporte para el aprendizaje colaborativo de programación en el nivel universitario inicial (resultados). XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires).
- RIPANI, M. F. (2017). Competencias de Educación Digital. Recuperado de https://www.argentina.gob.ar/sites/default/files/competencias_de_educacion_digital_1.pdf.
- ROSAS, M. V., ZÚÑIGA, M. E., FERNÁNDEZ, J., & GUERRERO, R. A. (2017). El pensamiento computacional en el ámbito universitario. XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires). Red de Universidades con Carreras en Informática (RedUNCI).
- ROSAS, M. V., ZÚÑIGA, M. E., FERNÁNDEZ, J., & GUERRERO, R. A. (2018). Pensando computacionalmente: ¿cómo, cuándo y dónde? y... ¿quiénes? XIII Congreso de Tecnología en Educación y Educación en Tecnología (Posadas, 2018).
- SALAZAR MESÍA, N., SANZ, C., & GORGA, G. (2016). Experiencia de enseñanza de programación con realidad aumentada. 213-220. Recuperado de <https://upcommons.upc.edu/handle/2117/90268>.
- SPIGARIOL, L. (2015). Desarrollo de un Lenguaje de Programación Orientado a Objetos con Fines Educativos: Una Experiencia de Aprendizaje para los Docentes. 3er Congreso Nacional de Ingeniería Informática/Sistemas de Información, 1229-1237. Buenos Aires, Argentina.
- TORRES, V., GARCÍA, A., LAMAS, R., & PÉREZ IBARRA, M. (2017). Los Videos Educativos como Herramientas para la Construcción del Conocimiento. Educación en Ingeniería, 1028-1034. Santa Fe, Argentina.
- VERA, P. M., MORENO, E. J., RODRÍGUEZ, R. A., VÁZQUEZ, M. C., VALLÉS, F. E., & CESCÓN, J. G. (2017). Gamificación en el ámbito universitario. 1230-1234. Recuperado de <http://hdl.handle.net/10915/62917>.
- VERA, P. M., RODRÍGUEZ, R. A., & MORENO, E. J. (2018). Gamificando la práctica de programación en el ámbito universitario. Campus Virtuales, 7(2), 55-68.

- WING, J. (2017). Computational Thinking's Influence on Research and Education for all. *Italian Journal on Educational Technology*, 25(2), 7-14.
- WING, J. M. (2006). Computational Thinking. *Commun. ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>.
- ZABALA, G., CERRATO, L. P., BLANCO, S., MORÁN, R., & TERAGNI, M. (2016). Minecraft Programable: Una herramienta para aprender programación en nivel medio. *Virtualidad, Educación y Ciencia*, 7(12), 113-124.
- ZAPATA-ROS, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. *RED. Revista de educación a distancia*, (46), 1–47.
- ZURITA, R., FUENTE, J. DE LA, BUCAREY, M., BONET, D., CASTILLO, R. DEL, RODRÍGUEZ, J.; CECCHI, L. (2017). Múltiplo N6 max adaptado para mejorar las posibilidades de aprender a programar. Presentado en XII Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET, La Matanza 2017). Recuperado de <http://sedici.unlp.edu.ar/handle/10915/63456>.